# Conclusão

Essa sessão tem como objetivo, partindo dos dados dos testes da páginas anteriores, definir quais implementações são as mais desejáveis para realizar a fatoração QR.

Para os exemplos usados, tivemos melhoria significativa entre `qr_GS` e `qr_GSM`. Acrescentar mais uma etapa, o pivoteamento parcial não se mostrou muito útil, pode ser que eu não tenha testado com um exemplo conveniente, mas não me parece o caso.

Quanto à acurácia e ortogonalidade, não percebi nenhuma diferença significativa entre usar uma função baseada no método de Gram-Schmidt ou Householder.

Para lidar com matrizes má condicionadas, as funções `qr_GSM`, `qr_GSP`, `qr_House_1` e `qr_House_2` estão equiparadas, a diferença entre elas não é tão relevante, pois quando grande, a diferença fica na escala de $10^2$ ou $10^3$.

Por fim, ressalto as funções preferíveis em cada caso:

Se $A \in \mathbb{M}^{m \times n}$, com $m \geq n$:

É recomendado o uso de `qr_GSM`, afinal é o caso mais simples dentre os que têm o seu nível de precisão.

Se $A \in \mathbb{M}^{m \times n}$, com $m < n$:

Prefira o uso de `qr_House_2`, pois esse caso, diferente de vários outros, é capaz de retorna uma fatoração funcional. Para o outro caso ($m \geq n$), essa versão faz com que o resultado seja o mesmo que o do método de GS. Para o cenário em questão, essa fatoração se parece com a fatoração reotrnada por `qr_House_1` para matrizes nas quais $m \geq n$